FACULDADE DE TECNOLOGIA DE AMERICANA

Curso Superior de Tecnologia em Segurança da Informação

Leonardo Rodrigues Ribeiro

Resumo:

Implementação do algoritmo quântico Deutsch-Jozsa em linguagem funcional e no simulador IBM Q Experience

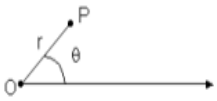
Americana, SP

2023

Definições

As definições matemáticas de álgebra linear são subdividas em 11 partes, sendo elas, espaço vetorial real, onde em um conjunto V, não vazio, está definido as operações de adição e multiplicação por escala, subespaço vetorial, onde considerando que S, um subconjunto não vazio do espaço vetorial V, seja um espaço vetorial em relação à adição e à multiplicação por escalar, combinação linear de vetores onde sejam os vetores, $v1, v2 \ldots vN$ do espaço vetorial V e os escalares $a1, a2 \ldots aN$. Qualquer V pertence V da forma $v = v1a1 + v2a2 + \ldots$ e uma combinação linear dos vetores $v1, v2$ ...

Produto interno em espaço vetorial, se define como produto escalar (ou interno) no espaço vetorial V uma aplicação de $V \times V$ em que a todo par de vetores $V \times V$ associa um número real. Coordenadas Polares, sistema de coordenadas bidimensional que descreve um ponto no espaço como um ângulo de rotação ao redor da origem e um raio a partir dela, sendo representada pelas seguintes coordenadas:



Números complexos que são elementos que habitam o conjunto , onde existe possibilidade de raízes negativas

Forma algébrica e Representação geométrica:

$$z = a + bi \ (a \in R, \ b \in R \ e \ i \ elevado \ a \ 2) = 1$$

A é a parte real de z, b é a parte imaginaria de z e i é a unidade imaginaria sendo o i o fator que possibilita a existência da raiz quadrada negativa. Podemos

Lei de Moore:

O físico estadunidense Gordon Moore estimou em 1965 que a cada 18 meses a capacidade de processamento iria aumentar em 100% enquanto seu custo permaneceria constante, posteriormente esta profecia provou-se real e passou a ser conhecida como Lei de Moore.

Notação de Dirac ou Bra-ket:

A notação de Dirac ou Bra-ket é a mais usada para descrever sistemas mecânicos quânticos e registradores quânticos.

Espaço de Hilbert:

Vetor espacial com um produto interno e uma regra definida pelo produto interno.

Conceitos de computação quântica

Iniciamos o conceito de computação quântica com a sobreposição de estado, estudado por Nielsen e Chuang foi possível entender o conceito de qubits (bit quântico), pois enquanto no clássico ele só pode ser 0 ou 1, no quântico ele pode apresentar os dois estados ao mesmo, sendo medido via probabilidade usando a fórmula: $\frac{1}{\sqrt{2}}|0 + \frac{1}{\sqrt{2}}|1>$.

Esfera de Bloch

Para ajudar na compreensão da Esfera de Bloch, local geométrico dele, utilizando apenas de Transformadores Unitários (T), que mudam a dimensão desse qubit sem mudar sua estrutura tendo com seu exemplo mais famoso a seguinte equação: $|א\rangle = e^{i\psi}[cos(\mathcal{E})|0\rangle + e^{i\varphi} sin(\mathcal{E})|1\rangle]$

Também é utilizado portas na computação quântica, mas diferente da computação clássica que é usado porta logica, na quântica utilizasse portas quânticas pois as operações de circuitos quânticos são reversíveis, sendo assim é necessário que o número de qubits da entrada seja igual ao número de qubits de saída. Logo abaixo veremos as principais portas logicas existentes:

Operadores de Pauli

Três das principais portas quânticas são conhecidas como os operadores de Pauli, e recebem a notação X, Y e Z.

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \qquad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

Portas de Fase

São responsáveis por realizar alterações na fase do qubit, possuindo quatro variações, $S, ‘, T$.

- A porta S, realiza a rotação de $\pi/2$ sendo equivalente a $\sqrt{Z}$, uma vez que Z realiza uma reação de π.
- A porta ‘ é a matriz S adjunta, realiza uma rotação de - $(\pi¦2)$.
- A porta T realiza uma rotação de $(\pi¦4)$ e é equivalente a $\sqrt{S}$ uma vez que S realiza uma rotação de $(\pi¦2)$.

Porta de Hadamard

É uma das portas quânticas mais úteis, pois é responsável por levar um estado a uma superposição.

Representação matemática: $T = \begin{pmatrix} 1 & 0 \\ 0 & 1+i \\ 1 & \sqrt{2} \\ & 0 \end{pmatrix}$

Algoritmo de Deutsch-Jozsa

Descreve um problema onde é necessário definir se uma função $f(x)$ é constante ou balanceada, e caso seja constante, todos resultados serão iguais a 0 ou iguais a 1.

Paradigmas de programação

A programação funcional decompõe o problema em uma série de funções, que recebem os inputs e produzem output, sem realizar mudança de estado internamente. Haskell é um exemplo de linguagem funcional muito conhecida e há linguagens multi-paradigmas, é possível escrever programas que são procedurais, orientadas a objeto ou funcionais, como é o caso de Python que possui uma sintaxe limpa e reputação de produtividade.

O IBM Quantum Experience permite que seus usuários se conectem aos processadores quânticos através do IBM Cloud, executem algoritmos e programas e apresenta um manual completo de uso, para iniciantes ou usuários avançados podendo acontecer através de modelo de circuitos ou linguagem de programação utilizando API Pyhton.

Haskell

Um programa funcional em Haskell consiste em uma série de definições, associando o nome ao valor de um tipo específico, basicamente funciona da seguinte maneira:

quadrado $n = n * n$

Linguagem Quipper

Quipper é uma linguagem de programação funcional integrada para Computação Quântica sendo baseada em Haskell, portanto pode ser considerada como uma série de data types, combinadores e bibliotecas de funções Haskell. Programas em Quipper são executados em três fases, sendo elas, analisa o código fonte, recebe como inputs o código de objeto da fase anterior, execução do circuito.

Implementação do algoritmo deutsch-jozsa em quipper/haskell

$import\ Quipper$

$data\ Oracle = Oracle\ \{$

$qubit\_num :: Int,$

$function :: ([Qubit], Qubit) \rightarrow Circ\ ([Qubit], Qubit)$

$\}$

$deutsch\_jozsa\_circuit :: Oracle \rightarrow Circ\ [Bit]$

$deutsch\_jozsa\_circuit\ oracle = do$

$top\_qubits \leftarrow qinit\ (replicate\ (qubit\_num\ oracle)\ False)$

$bottom\_qubit \leftarrow qinit\ True$

$label\ (top\_qubit, bottom\_qubit)\ ("|0\rangle\ ", "|1\rangle\ ")$

$mapUnary\ hadamard\ top\_qubits$

$hadamard\_at\ bottom\_qubit$

$function\ oracle\ (top\_qubits, bottom\_qubit)$

```
mapUnary hadamard top_qubits

(top_qubits, bottom_qubit) ← measure (top_qubits, bottom_qubit)

main = print_generic Preview (deutsch_jozsa_circuit empty_oracle)

where

empty_oracle :: Oracle

empty_oracle = Oracle {

qubit_num = 5,

function = empty_oracle_function

}

empty_oracle_function:: ([Qubit], Qubit) → Circ ([Qubit], Qubit)

empty_oracle_function (ins, out) = named_gate "Oracle" (ins, out)

import qualified Data.Map as Map

import QuipperLib.Simulation

import System.Random

simulate :: Circ [Bit] → Bool

simulate oracle = and (map not (run_generic (mkStdGen 1) (1.0:: Float) oracle))

circuit :: (Circ [Bit] → Bool) → Oracle → IO ()

circuit run oracle = if run (deutsch_jozsa_circuit oracle)

then putStrLn "constant"

else putStrLn "balanced"

main = do

circuit simulate constant_oracle

circuit simulate balanced_oracle
```